

Computing rank of finite algebraic structures with limited nondeterminism

Jeffrey Finkelstein

Computer Science Department, Boston University

February 16, 2016

Abstract

The rank of a finite algebraic structure with a single binary operation is the minimum number of elements needed to express every other element under the closure of the operation. In the case of groups, the previous best algorithm for computing rank used polylogarithmic space. We reduce the best upper bounds on the complexity of computing rank for groups and for quasigroups. This paper proves that the rank problem for these algebraic structures can be verified by highly restricted models of computation given only very short certificates of correctness.

Specifically, we prove that the problem of deciding whether the rank of a finite quasigroup, given as a Cayley table, is smaller than a specified number is decidable by a circuit of depth $O(\log \log n)$ augmented with $O(\log^2 n)$ nondeterministic bits (the complexity class of problems decidable by such circuits is denoted $\beta_2\text{FOLL}$). Furthermore, if the quasigroup is a group, then the problem is also decidable by a Turing machine using $O(\log n)$ space and $O(\log^2 n)$ bits of nondeterminism with the ability to read the nondeterministic bits multiple times (the complexity class for problems like this is denoted $\beta_2\text{L}$). Finally, we provide similar results for related problems on other algebraic structures and other kinds of rank. These new upper bounds are significant improvements, especially for groups. In general, the lens of limited nondeterminism provides an easy way to improve many simple algorithms, like the ones presented here, and we suspect it will be especially useful for other algebraic algorithms.

1 Introduction

An efficient algorithm computing the rank of a finite algebraic structure (that is, the minimum number of elements required to generate all other elements) benefits

Copyright 2014, 2015, 2016 Jeffrey Finkelstein (jeffreyf@bu.edu).

This document is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License, which is available at <https://creativecommons.org/licenses/by-sa/4.0/>. The L^AT_EX markup that generated this document can be downloaded from its website at <https://github.com/jfinkels/grouprank>. The markup is distributed under the same license.

mathematicians, who use numerical algebra systems for research, cryptographers, who rely on algebraic systems for proofs of security, and theoretical computer scientists, who seek to understand which problems can be solved in a particular model of computation. If the structure is, for example, a finite group, then we can represent this structure in one of two reasonable ways. First, we can represent it as a subset of elements along with a set of equality relations demonstrating how the group operation behaves (known as a group presentation). Second, we can represent it as a table of values for the binary operation under each pair of input elements (known as a Cayley table or multiplication table). These representations offer a tradeoff between representation size and the complexity of deciding properties of the group: the latter representation may be exponentially larger than the former, so an efficient algorithm for the latter may not necessarily be efficient for the former.

Consider the situation when the algebraic structure is the finite cyclic group of order n . A natural presentation of this group is $\langle a \mid a^n = 1 \rangle$. Since each element in this group can be represented by $O(\log n)$ bits, the total size of this representation is $O(\log n)$ bits. In contrast, the Cayley table for this group requires $O(n^2 \log n)$ bits. Thus, in certain cases, if m represents the size of the input, an algorithm running in time $f(m)$ on inputs of the first form runs in time $O(f(\log m))$ on inputs of the second form. We can use this to our advantage to construct more efficient algorithms for algebraic problems.

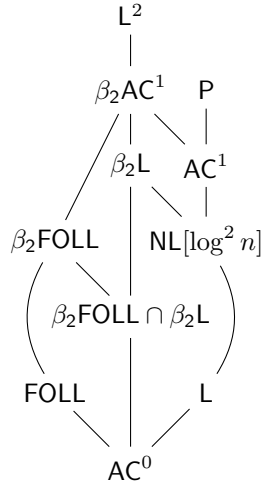
For quasigroups, the previous best algorithm for computing the rank requires polynomial time in addition to a polylogarithmic amount of nondeterministic bits. For groups, the previous best algorithm for computing the rank requires a polylogarithmic amount of space, which induces a superpolynomial time (hence, inefficient) algorithm. Only for certain classes of finite groups is there a polynomial-time algorithm. We improve the best upper bound on the complexity of the rank problem for quasigroups and groups by using an algorithm with limited nondeterminism. This paper proves that with short certificates of correctness, the rank problem for quasigroups and groups can be verified by highly restricted models of computation, and demonstrates how the same strategy can be applied to other algebraic structures.

We prove that the problem of deciding whether the rank of a finite quasigroup, given as a Cayley table, is smaller than a specified number is decidable by a circuit of depth $O(\log \log n)$ augmented with $O(\log^2 n)$ nondeterministic bits (the complexity class of problems decidable by such circuits is denoted $\beta_2\text{FOLL}$). Furthermore, if the quasigroup is a group, then the problem is also decidable by a Turing machine using $O(\log n)$ space and $O(\log^2 n)$ bits of nondeterminism with *two-way* read access to the nondeterministic bits (the complexity class for problems like this is denoted $\beta_2\text{L}$). The general strategy is to reduce the problem of computing rank to the problem of computing membership; we compute the rank of a group by guessing a small set of candidate generators, then deciding whether each other element in the group can be generated from that set. For the sake of completeness, we show how this strategy applies to semigroups and magmas in general, though these results are less interesting because those algebraic structures lack the small generating sets that quasigroups have. We show that

Table 1: We improve algorithms for computing rank of finite algebraic structures.

	old	new
magma	NP	NP
semigroup	NP	NP
quasigroup	β_2P	β_2FOLL
group	L^2	$\beta_2FOLL \cap \beta_2L$
ring	NP	β_2AC^1

Figure 1: Hierarchies of complexity classes with limited nondeterminism can circumvent common deterministic complexity classes.



the rank problem for rings is decidable by a circuit of depth $O(\log n)$ augmented with $O(\log^2 n)$ nondeterministic bits (a class denoted β_2AC^1). Finally, we show how this technique applies to other notions of rank. Table 1 summarizes these improvements, and Figure 1 demonstrates graphically why these improvements are so significant. (We could not find an explicit upper bound for magmas, semigroups, or rings, but other papers imply that the problems are in NP.)

These are improvements on the previous best upper bounds for these problems. Previously, the best upper bound for computing the rank of a quasigroup given as a Cayley table was β_2P [21, Section 5] and for groups, L^2 [17] (see also [1, Proposition 6] for a brief description of the algorithm). Our results are an improvement because $\beta_2FOLL \subseteq \beta_2P$ and $\beta_2L \subseteq L^2$. Our algorithm is still not in P, so these algorithms do not supercede [1, Theorem 7], which gives a polynomial-time algorithm that computes the rank of a nilpotent group. Furthermore, the

relationship between FOLL and L remains unknown (the best inclusion known is the uninteresting inclusion $\text{FOLL} \subseteq \text{AC}^1$), so the relationship between $\beta_2\text{FOLL}$ and $\beta_2\text{L}$ is unknown as well. Finally, contrast the complexity of the rank problem for groups with the complexity of computing the rank of a subgroup of a free group: the latter problem is P -complete, so is not even in NC unless $\text{NC} = \text{P}$ [3, Theorem 4.9] (see also [10, Problem A.8.11]).

Using limited nondeterminism and restrictive models of computation as verifiers may also be useful in examining other problems. The limited nondeterminism lens specifically suggests some opportunities for further research in computational algebra, though it has seen some recent success in other subfields of theoretical computer science (see [8], for example). Here are some avenues for future research.

- Is computing the rank of a quasigroup also in $\beta_2\text{L}$?
- Is the group rank problem in a smaller complexity class, one contained in both $\beta_2\text{FOLL}$ and $\beta_2\text{L}$? What is the largest complexity class we can find that is in $\text{FOLL} \cap \text{L}$? This would likely improve all the results in [5].
- Is there a reduction between the problem of computing the rank of a quasigroup and the problem of deciding whether two quasigroups are isomorphic?
- Is the problem of computing the shortest generating sequence for a quasigroup strictly more difficult than the problem of computing the rank of a quasigroup?
- The complexity of group problems, for example, varies based on the succinctness of the representation of the input. In this paper we show that the rank problem is quite easy when the input is given its least succinct representation, the full Cayley table. On the other hand, in the most succinct representation, the group presentation (a set of generators for the group along with relations among the generators), many problems become very difficult, or even undecidable if the group is infinite. For representations of intermediate succinctness, for example a circuit that outputs the entries of the Cayley table, how difficult is the rank problem?

2 Preliminaries

Here, $\log n$ denotes the base two logarithm of n , for any natural number n .

2.1 Complexity

Here is a brief summary of the definitions of the complexity classes that appear in this paper.

- L is the class of languages decidable by a deterministic Turing machine that uses $O(\log n)$ space on inputs of length n . L^2 is similar, but with $O(\log^2 n)$ space.
- NL is the class of languages decidable by a nondeterministic Turing machine that uses $O(\log n)$ space. Equivalently, this is the class of languages L for which there is a deterministic Turing machine with a two-way read-only tape for the input string, a one-way read-only tape for the nondeterministic bits, and a two-way read-write work tape in which only $O(\log n)$ cells are used, such that $x \in L$ if and only if there is a binary string w of polynomial length such that the machine accepts on input x and nondeterministic bits w .
- $NL[\log^2 n]$ is the subclass of NL in which the length of w is bounded by $O(\log^2 n)$.
- $\beta_2 L$ is the superclass of $NL[\log^2 n]$ in which the machine has two-way access to the tape containing the nondeterministic bits. Equivalently, this is the class of languages L such that there is a language $L' \in L$ such that $x \in L$ if and only if there is a binary string w of length $O(\log^2 n)$ such that $(x, w) \in L'$.
- $FOLL$ is the class of languages decidable by a L -uniform family of circuits with polynomial size, unbounded fan-in, and $O(\log \log n)$ depth. $\beta_2 FOLL$ is the class of languages decidable by $FOLL$ circuits that have been augmented with $O(\log^2 n)$ nondeterministic bits (gates with no inputs and one output).
- AC^0 and $\beta_2 AC^0$ are the restrictions of $FOLL$ and $\beta_2 FOLL$, respectively, to depth $O(1)$. NAC^0 allows a polynomial number of nondeterministic bits.

In general, the class $\beta_2 \mathcal{C}$ is the class of languages decidable by \mathcal{C} machines augmented with $O(\log^2 n)$ bits of nondeterminism, or equivalently, the class of languages verifiable by \mathcal{C} machines when given a certificate of length $O(\log^2 n)$.

If L_1 and L_2 are languages, there is a *logarithmic space many-one reduction* from L_1 to L_2 , denoted $L_1 \leq_m^L L_2$, if there is a function f such that f is computable in logarithmic space and $x \in L_1$ if and only if $f(x) \in L_2$. There is a $\beta_2 AC^0$ *conjunctive truth-table reduction* from L_1 to L_2 , denoted $L_1 \leq_{ctt}^{\beta_2 AC^0} L_2$, if there is a function f and a polynomial p such that

- f is computable in AC^0 ,
- $x \in L_1$ if and only if there is a w of length $O(\log^2 n)$ such that $\bigwedge_{i=1}^{p(n)} y_i \in L_2$, where $f(x, w) = (y_1, \dots, y_{p(n)})$.

A NAC^0 conjunctive truth-table reduction is the generalization in which f receives a witness of length polynomial in n , instead of polylogarithmic in n . A reduction of this form is really a nondeterministic polynomial-time conjunctive truth-table reduction, since $NAC^0 = NP$.

Lemma 2.1. *Suppose L_1 and L_2 are languages.*

1. *If $L_1 \leq_{ctt}^{\text{NAC}^0} L_2$ and L_2 is in \mathbf{P} , then L_1 is in \mathbf{NP} .*
2. *If $L_1 \leq_{ctt}^{\beta_2 \text{AC}^0} L_2$ and L_2 is in \mathbf{FOLL} , then L_1 is in $\beta_2 \mathbf{FOLL}$.*
3. *If $L_1 \leq_{ctt}^{\beta_2 \text{AC}^0} L_2$ and L_2 is in \mathbf{L} , then L_1 is in $\beta_2 \mathbf{L}$.*

Proof. In each case, let f denote the reduction, M_2 denote the machine that decides L_2 , and $q(n)$ denote the polynomial that bounds the number of outputs produced by f . We construct a nondeterministic machine M_1 of the appropriate type as follows on input x of length n . Nondeterministically choose a string w of the appropriate length (polynomial or polylogarithmic), simulate $f(x, w)$, then run M_2 on each y_i , where $f(x, w) = (y_1, \dots, y_{q(n)})$. The machine M_1 accepts if and only if each of the simulations of M_2 accepts. The correctness of M_1 follows from the correctness of f and M_2 . The only remaining issue is the complexity of M_1 .

In the first case, the \mathbf{NP} machine M_1 , after choosing its nondeterministic bits, can simulate f in polynomial time and can simulate a polynomial number of instances of M_2 in polynomial time.

For the last two cases, we use the fact that $\beta_2 \text{AC}^0 \subseteq (\beta_2 \mathbf{FOLL} \cap \beta_2 \mathbf{L})$. If L_2 is in \mathbf{FOLL} , we define M_1 to be the circuit

$$M_1(x, w) = \bigwedge_{i=1}^{q(n)} M_2(y_i),$$

where n is the length of x , the string w is the nondeterministic string of length $O(\log^2 n)$, and $q(n)$ is the polynomial bounding the number of outputs of f on inputs of length n . The depth of the M_1 circuit is the depth of f plus the depth of M_2 , which is $O(1) + O(\log \log n)$, or simply $O(\log \log n)$. The number of nondeterministic bits required by M_1 is the same as the number of nondeterministic bits required by f , which is $O(\log^2 n)$. The circuit is polynomial in size because f is polynomial in size, M_2 is polynomial in size, and there are a polynomial number of parallel instances of the circuit M_2 . Thus M_1 is in $\beta_2 \mathbf{FOLL}$.

The proof is similar if L_2 is in \mathbf{L} . The only difference is that instead of a circuit computing the conjunction of $q(n)$ bits, we loop over each y_i and check if each one causes M_2 to accept. Since there are a polynomial number of them, indexing them requires only logarithmic space. We also require the fact that logarithmic space computable functions compose. \square

Conondeterministic reductions yield similar closures.

Finally, if L is a language and F is a function, there is a *nonadaptive* AC^0 *Turing reduction* from L to F if there is an AC^0 function g and an AC^0 circuit C such that $x \in L$ if and only if $C(x, F(y_1), \dots, F(y_m)) = 1$, where (y_1, \dots, y_m) is the output of $g(x)$ and m is bounded by a polynomial in $|x|$. The function g is called the *generator of the reduction* and the circuit C is called the *evaluator of the reduction*.

2.2 Algebra

A *magma* is a set G with a binary operation \cdot that is closed on G . Unless otherwise stated, we will only consider *finite magmas*, in which G is a finite set. The *Cayley table* of a magma with n elements is the $n \times n$ table whose rows and columns are indexed by the elements of G and where entry (a, b) has value c if $a \cdot b = c$. If the binary operation is associative, the magma is called a *semigroup*. A semigroup with a unique identity element is called a *monoid*. If the binary operation has the property that for each a and b in G there are unique elements x and y in G such that $a \cdot x = b$ and $y \cdot a = b$, the magma is called a *quasigroup*. (In other words, each quasigroup element appears exactly once in each row and each column of the Cayley table of G , or the Cayley table is a *Latin square*.) A quasigroup with at least one identity element is called a *loop*. If a quasigroup is nonempty and associative, then it is a *group*. Alternately, if a semigroup has an identity and inverses, then it is a group.

Example 2.2. The smallest nonempty quasigroup that is not also a group has three elements, $\{a, b, c\}$. Its Cayley table is

\cdot	a	b	c
a	a	b	c
b	c	a	b
c	b	c	a

Examining the table reveals that there is exactly one of each quasigroup element in each row and column. This quasigroup is not associative because $b \cdot (a \cdot b) = b \cdot b = a$ but $(b \cdot a) \cdot b = c \cdot b = c$. Also, it has a left identity, a , but no right identity. \square

Example 2.3. The *right zero semigroup* is the semigroup in which each element is a right zero. Its Cayley table is

\cdot	a	b	c
a	a	b	c
b	a	b	c
c	a	b	c

The associativity of this semigroup can be determined by examining all possible triples (x, y, z) in G^3 and checking that $(x \cdot y) \cdot z = x \cdot (y \cdot z)$. Each element of this semigroup is a left identity and a right zero, but there are no right identities, so it is not a group.

Unlike for the Latin square property in the previous example, there is no obvious way to tell whether a binary operation is associative simply by scanning the rows and columns. In other words, given only its Cayley table, determining whether a magma is a quasigroup *seems* easier than determining whether a magma is a semigroup. However, there is a polynomial time algorithm, attributed to F. W. Light, for deciding whether a magma is associative; it is simply the naïve algorithmic implementation of the associativity condition. \square

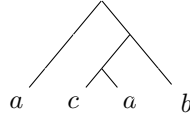
Example 2.4. There is a unique (up to isomorphism) group on three elements $\{a, b, c\}$. Its Cayley table is

\cdot	a	b	c
a	a	b	c
b	b	c	a
c	c	a	b

This group is the group of integers under addition modulo three. It is both a semigroup and a quasigroup. \square

A *parenthesization* P of a sequence of magma elements (g_1, \dots, g_k) is a binary tree that has the magma elements as its leaves (in the order indicated by the sequence). The *parenthesized product* of a sequence of magma elements (g_1, \dots, g_k) with parenthesization P , denoted $P(g_1, \dots, g_k)$, is the element that results from performing the magma operation in the order indicated by the parenthesization. If S is a subset of magma elements, the *submagma* generated by S , denoted $\langle S \rangle$, is the closure of S under the magma operation and under any parenthesization. (For semigroups, and hence for groups, the operation is associative, so the parenthesization is superfluous.)

Example 2.5. Consider (a, c, a, b) , a sequence of four elements from the quasigroup defined in Example 2.2. One parenthesization of this sequence is



which corresponds to the parenthesized product $a \cdot ((c \cdot a) \cdot b)$. According to the Cayley table, this product equals a . \square

Lemma 2.6. For any quasigroup on n elements given as a Cayley table, any sequence (g_0, \dots, g_k) , and any parenthesization P of depth d on that sequence, the parenthesized product $P(g_0, \dots, g_k)$ can be computed by an L -uniform family of unbounded fan-in circuits with size $O(kn^2 \log n)$ and depth $O(d)$.

Proof. How does a circuit access and use a Cayley table for a quasigroup?¹ One way for a circuit to compute the product of two quasigroup elements using the Cayley table is via a multiplexer. In the multiplexer, each input has $O(\log n)$ bits, since each quasigroup element can be represented with $O(\log n)$ bits and each input is a pair of quasigroup elements. A multiplexer that selects from n^2 inputs, each of length $O(\log n)$, can be implemented by an *unbounded fan-in* circuit of depth $O(1)$ and size $O(n^2 \log n)$. \square

¹We avoid representing problems using first-order logic, as in the original definition of **FOLL** from [5], though the logic definition may provide a more natural representation of this sort of information.

2.3 Independent and generating sets

An element x of a subset S of a magma is *independent with respect to S* if x is not in $\langle S \setminus \{x\} \rangle$. A subset S is *independent* if each element of S is independent with respect to S .

Example 2.7. In the finite group C_6 , the cyclic group on six elements, the set $\{g^2, g^3\}$ is independent but the set $\{g^2, g^4\}$ is not independent. \square

The dual notion of an independent set is that of a generating set. If $\langle S \rangle = G$, then S is called a *generating set for G* . The *rank of a magma G* , denoted $\text{rank}(G)$, is the minimum cardinality of a generating set. This terminology extends to semigroups and groups as well.

Example 2.8. Consider the right zero semigroup on n elements, a generalization of Example 2.3. In this semigroup, call it G , we have $x \cdot y = y$ for each x and y in G . The rank of this semigroup must be n . Assume for the sake of producing a contradiction that the rank is strictly less than n . Thus there is an element z not in the generating set such that $x_1 \cdots x_n = z$, where each x_i is an element of the generating set. This is a contradiction with the fact that $x_1 \cdots x_n = x_n$, since x_n is a right zero. Therefore the rank of the semigroup must be n . \square

Example 2.9. Consider the elementary abelian 2-group, $(\mathbb{Z}/2\mathbb{Z})^k$, for some positive integer k . Let n denote the order of this group, so $n = 2^k$. The minimum generating set for this group is $\{e_1, \dots, e_k\}$, where e_i is the k -tuple with a one in the i th position and a zero in each other position (if we consider the group as a vector space, e_i is the standard basis vector). Thus the group has a minimum generating set of size k , which is $\log n$. \square

For quasigroups, we consider a slightly more specific notion of “generating.” If (g_0, \dots, g_k) is a finite sequence of quasigroup elements denoted S and P is a parenthesization of that sequence, then the *cube of S with respect to P* , denoted $\text{cube}_P(S)$, is defined

$$\text{cube}_P(S) = \{P(g_0, g_1^{\epsilon_1}, \dots, g_k^{\epsilon_k}) \mid \epsilon_i \in \{0, 1\} \text{ for each } i\},$$

where $g_i^{\epsilon_i}$ denotes g_i if $\epsilon_i = 1$ and the empty word if $\epsilon_i = 0$. The element g_0 has no exponent because the empty word by itself is not an element in a quasigroup. (This is called a “cube” because each vertex of the k -dimensional Boolean hypercube, when interpreted as a binary string $\epsilon_1 \cdots \epsilon_k$, yields a quasigroup element.) If $\text{cube}_P(S) = G$, then g is called a *cube generating sequence of size $k+1$ for the quasigroup G* . The *rank of a quasigroup G* , denoted $\text{rank}(G)$, is the minimum size of a cube generating sequence.² Contrast the rank of a quasigroup with the rank of a semigroup: the former is the size of a sequence, the latter the size of a set.

² This is a nonstandard definition of “rank” for quasigroups. Elsewhere, the rank of a quasigroup is the number of blocks in the partition of the quasigroup into conjugacy classes according to the action of the quasigroup on itself.

Ideally, we would like the notion of rank to be identical for each algebraic structure. If a quasigroup has a cube generating sequence of size at most k , then it has a generating set of size at most k , specifically the set of distinct elements from the sequence. However, we conjecture that for sufficiently large sizes, there is a quasigroup that has a generating set of size strictly less than the size of its minimum cube generating sequence. If this is incorrect, that is if a small generating set implies a small cube generating sequence, we could use the same definition of rank for all our algebraic structures, simplifying our proofs.

Quasigroups have small cube generating sequences, and groups have small generating sets. As of this publication, upper bounds on the size of generating sets for semigroups remain the subject of research [9], although in general, some semigroups of order n have rank n (see Example 2.8). Magmas have even less structure than semigroups, and hence lack a meaningful upper bound as well.

An upper bound for the minimum size of a generating set for quasigroups can be proven by the probabilistic method.

Lemma 2.10 ([6, Theorem 3.3]). *Each finite quasigroup with n elements has a cube generating sequence of size $O(\log n)$ with a parenthesization of depth $O(\log \log n)$.*

Since a group is a quasigroup, and since a cube generating sequence induces a generating set, the same upper bound can be applied to groups. However, a more specific (and constructive) upper bound can be proven inductively by considering cosets of increasing size. In fact, we can prove a more general lemma bounding the size of independent sets that we will use later as well.

Lemma 2.11. *Suppose G is a finite magma of order n such that for each submagma H and each $x \in G \setminus H$, the cosets xH and H are disjoint. Each independent subset of G has size at most $\log n$.*

Proof. Let S be an independent subset of G of cardinality m . Define the sequence of sets H_0, \dots, H_m by

$$\begin{aligned} H_0 &= \emptyset \\ H_i &= \langle H_{i-1} \cup \{x_i\} \rangle, \end{aligned}$$

where x_i is the i th element of S (in an arbitrary ordering). Since each x_i is taken from S , an independent set, $x_i \notin \langle H_{i-1} \rangle$. For each $i \in \{1, \dots, m\}$, we have $x_i H_{i-1}$ and H_{i-1} are disjoint by hypothesis. Thus $|H_i| \geq 2|H_{i-1}|$, and by induction $|H_m| \geq 2^m$. We conclude that $m \leq \log |H_m| \leq \log |G| = \log n$. \square

The disjointness requirement in the previous lemma is fairly strict. Neither semigroups (the zero semigroup, for example) nor loops necessarily satisfy it. Still, a finite group satisfies the hypothesis in the previous lemma, so each independent set of a finite group has size at most $\log n$. Furthermore, each finite group has an independent *generating* set.

Lemma 2.12. *Each finite group has an independent generating set, and specifically its minimum cardinality generating set is independent.*

Proof. Let S be a generating set of minimum cardinality among all generating sets. If S is independent, we are done. If S is not independent, then there is an element x in S such that $x \in \langle S \setminus \{x\} \rangle$. But then $\langle S \rangle = \langle S \setminus \{x\} \rangle$, a contradiction with the minimality of S . Therefore the minimum cardinality generating set is independent. \square

Combining the previous two lemmas we conclude that the minimum cardinality of a generating set of a finite group is at most $\log n$; Example 2.9 gives a group that achieves the upper bound.

Lemma 2.13. *If G is a finite group of order n then the minimum cardinality of a generating set is at most $\log n$, with equality when the group is a finite elementary abelian 2-group.*

3 Computation of submagma membership

As stated in the introduction, computing the rank of a magma reduces to the problem of deciding submagma membership. Thus, in order to determine an upper bound on the complexity of computing the rank, we can determine the complexity of deciding membership. Fortunately, the membership problem for a submagma when the magma is given as its Cayley table is a well-studied problem. This section reviews the complexity for the membership problem for magmas, semigroups, quasigroups, and groups.

We recall that for magmas the membership problem, given the Cayley table, is in NP and for semigroups NL. We prove that for quasigroups the problem is in FOLL and for groups L. These upper bounds allow us to prove upper bounds on the rank problem in the next section. If future work reveals more efficient algorithms for the membership problem for quasigroups or groups, we can provide improved algorithms for computing the rank of these algebraic structures.

The SUBMAGMA MEMBERSHIP problem is defined as follows. The inputs are a magma G given as a Cayley table, a magma element h , and a finite set of magma elements S . The problem is to decide whether $h \in \langle S \rangle$.

Lemma 3.1 ([14, Corollary 9]). SUBMAGMA MEMBERSHIP is P-complete.

The SUBSEMGROUP MEMBERSHIP problem is defined as follows. The inputs are a semigroup G given as a Cayley table, a semigroup element h , and a finite set S of semigroup elements. The problem is to decide whether $h \in \langle S \rangle$.

Lemma 3.2 ([15]). SUBSEMGROUP MEMBERSHIP is NL-complete.

The CUBE MEMBERSHIP problem is defined as follows. The inputs are a quasigroup G given as a Cayley table, a quasigroup element h , a finite sequence of quasigroup elements S , and a parenthesization P for that sequence. The problem is to decide whether $h \in \text{cube}_P(S)$.

Lemma 3.3 (Implicit in [6, Theorem 3.4]). CUBE MEMBERSHIP is decidable by an L-uniform family of unbounded fan-in circuits with size $O(2^k k n^2 \log n)$ and

depth $O(d)$, where n is the order of the quasigroup, k is the size of the generating sequence, and d is the depth of the parenthesization.

In particular, if $k = O(\log n)$ and $d = O(\log \log n)$, then CUBE MEMBERSHIP is in FOLL.

Proof. The input to the circuit is the Cayley table for a quasigroup, a quasigroup element h , a generating sequence S , and a parenthesization P . Suppose $S = (g_0, \dots, g_k)$ for some positive integer k . Since the circuit needs to determine if h is in $\text{cube}_P(S)$, the circuit accepts if and only if there is some sequence of bits $(\epsilon_1, \dots, \epsilon_k)$ such that $h = P(g_0, g_1^{\epsilon_1}, \dots, g_k^{\epsilon_k})$. Thus the circuit consists of 2^k subcircuits joined to a single OR gate, each subcircuit deciding whether one of the 2^k possible k -bit sequences $(\epsilon_1, \dots, \epsilon_k)$ produces h under the given parenthesization.

The subcircuit corresponding to binary sequence $(\epsilon_1, \dots, \epsilon_k)$ computes the parenthesized product $P(g_0, g_1^{\epsilon_1}, \dots, g_k^{\epsilon_k})$. Computing the parenthesized product can be implemented in $O(kn^2 \log n)$ size and $O(d)$ depth by Lemma 2.6. Comparing the element produced this way to the element h can be done with a constant depth, $O(\log n)$ size equality comparison circuit.

We conclude that the overall size of the circuit is $O(2^k kn^2 \log n)$ and the overall depth of the circuit is $O(d)$. \square

Although the notion of cube generating sequence will give us a better upper bound for computing quasigroup rank, we prefer to consider the more natural notion of a generating set, as we did for magmas, semigroups, and quasigroups. Since we know from Lemma 2.10 that we need only consider candidate generating sets of size $O(\log n)$ and candidate parenthesizations of depth $O(\log \log n)$, we use a generalized form of the quasigroup membership problem that allows us to specify bounds on the generating set size and parenthesization depth.

The BOUNDED SUBQUASIGROUP MEMBERSHIP problem is defined as follows. The inputs are a quasigroup G given as a Cayley table, a quasigroup element h , a finite set S of semigroup elements, a positive integer k , and a positive integer d . The problem is to decide whether there is a sequence s in S^k and a parenthesization of depth d on k elements such that $h = P(s)$. (This is the definition of “ $h \in \langle S \rangle$ ”, but with specific size and depth bounds on the binary tree that generates h .) This problem should be at least as difficult as CUBE MEMBERSHIP: the former requires finding an appropriate sequence and parenthesization, whereas for the latter, they are fixed beforehand.

Lemma 3.4. BOUNDED SUBQUASIGROUP MEMBERSHIP is decidable by an L-uniform family of unbounded fan-in circuits with size $O(n^2 k \log n)$ and depth $O(d)$, using $O(k \log n)$ nondeterministic bits.

In particular, if $k = O(\log n)$ and $d = O(\log \log n)$, then BOUNDED SUBQUASIGROUP MEMBERSHIP is in $\beta_2\text{FOLL}$.

Proof. The algorithm is similar to that of Lemma 3.3, except now we must nondeterministically choose a sequence and parenthesization. The circuit nondeterministically chooses k elements of S and a parenthesization of k elements of

depth d , then accepts if and only if that parenthesized product is h . Choosing k elements, each of size $O(\log n)$, requires $O(k \log n)$ bits and choosing a parenthesization requires $O(k)$ bits, so the total number of nondeterministic bits required is $O(k \log n)$. By Lemma 2.6, computing the parenthesized product requires a circuit of size $O(kn^2 \log n)$ and depth $O(d)$. The final equality comparison requires size $O(\log n)$ and depth $O(1)$. We conclude that the overall size of the circuit is $O(kn^2 \log n)$ and the overall depth of the circuit is $O(d)$. \square

The SUBGROUP MEMBERSHIP problem is defined as follows. The inputs are a group G given as a Cayley table, a group element h , and a finite set S of group elements. The problem is to decide whether $h \in \langle S \rangle$.

Lemma 3.5. SUBGROUP MEMBERSHIP is in L.

Proof. The problem is in SL by a reduction to UNDIRECTED PATH [4, Section 3], and SL = L [22]. \square

4 Computation of magma rank

Computing submagma membership is where most of the work occurs. Now we need only reduce the rank problem to the membership problem. We do this via a truth-table reduction of relatively low complexity. This section uses these reductions and the results of the previous section to prove the upper bounds on the rank problem as advertised in the introduction.

Theorem 4.2 proves that for semigroups and magmas, the rank problem is in NP (which was already known), for quasigroups $\beta_2\text{FOLL}$, and for groups $\beta_2\text{FOLL} \cap \beta_2\text{L}$. This means that for groups and quasigroups, the problem can be verified quickly in parallel given a very short witness. We conjecture that for magmas and semigroups, the problems are hard for their respective complexity classes.

The MAGMA RANK problem is defined as follows. Given the Cayley table of a magma and an integer k in unary, decide whether the rank of the magma is k or less. The restrictions of this problem to quasigroups, semigroups, and groups, respectively, are defined similarly. The integer k is given in unary in order to facilitate the construction of uniform circuit families that decide the problem; since the size of the Cayley table is $n^2 \log n$ and k is always at most n , encoding the integer in unary does not cause an exponential increase in the size of the input to the problems.

The reductions in the following lemma are implicit in [6, Theorem 3.4]. That theorem demonstrates a $\beta_2\text{FOLL}$ algorithm for deciding whether two quasigroups are isomorphic, and the first part of that algorithm determines whether a given sequence of quasigroup elements with a parenthesization is a cube generating sequence.

Lemma 4.1.

1. MAGMA RANK $\leq_{ctt}^{\text{NAC}^0}$ SUBMAGMA MEMBERSHIP.

2. SEMIGROUP RANK $\leq_{ctt}^{\text{NAC}^0}$ SUBSEMIGROUP MEMBERSHIP.
3. QUASIGROUP RANK $\leq_{ctt}^{\beta_2 \text{AC}^0}$ CUBE MEMBERSHIP.
4. QUASIGROUP RANK $\leq_{ctt}^{\beta_2 \text{AC}^0}$ BOUNDED SUBQUASIGROUP MEMBERSHIP.
5. GROUP RANK $\leq_{ctt}^{\beta_2 \text{AC}^0}$ SUBGROUP MEMBERSHIP.

Proof. First, consider the problem for magmas. Unlike for quasigroups and groups (Lemma 2.10 and Lemma 2.13), we have no general upper bound on the minimum size of a generating set for magmas. Thus, the best we can do is nondeterministically choose a set of k generators and determine if that set generates the magma, where k can be as large as the number of elements in the magma.

Let g_1, \dots, g_n denote the elements of a magma. The reduction proceeds as follows. On input (G, k) , where G is a magma on n elements given as its Cayley table and k is a positive integer given in unary, nondeterministically choose a sequence S of k magma elements. Output $((G, g_1, S), \dots, (G, g_n, S))$.

Since each magma element can be represented by $O(\log n)$ bits, the number of nondeterministic bits used is $O(n \log n)$. By definition of rank,

$$\text{rank}(G) \leq k \iff \bigwedge_{i=1}^n g_i \in \langle S \rangle,$$

so the reduction is a correct conjunctive truth-table reduction. For semigroups, we apply the exact same reduction.

For quasigroups, in the reduction to the cube membership problem, the only differences are that we need to nondeterministically choose a parenthesization as well as a generating sequence, and that we have an upper bound on the size of the sequence and the parenthesization. By Lemma 2.10, it suffices to consider inputs to QUASIGROUP RANK in which k is in $O(\log n)$ and inputs to CUBE MEMBERSHIP in which P is of depth $O(\log \log n)$. The reduction therefore must nondeterministically choose a sequence S of k quasigroup elements and a parenthesization P of depth $O(\log \log n)$. The output of the reduction is $((G, g_1, S, P), \dots, (G, g_n, S, P))$. Now the number of nondeterministic bits used is $O(\log^2 n)$, since S is a set of $O(\log n)$ strings, each of length $O(\log n)$. By Lemma 2.10,

$$\text{rank}(G) \leq k \iff \bigwedge_{i=1}^n g_i \in \text{cube}_P(S),$$

so the reduction is a correct conjunctive truth-table reduction.

For the reduction to the bounded quasigroup membership problem, we still need $O(\log^2 n)$ bits to guess the generating set, but we can let $k = O(\log n)$ and $d = O(\log \log n)$, by Lemma 2.10. Thus the reduction outputs $((G, g_1, S, k, d), \dots, (G, g_n, S, k, d))$, and the correctness of the reduction follows from the fact that

$$\text{rank}(G) \leq k \iff \bigwedge_{i=1}^n g_i \in \langle S \rangle.$$

The proof for groups is again similar. Instead of Lemma 2.10, we invoke Lemma 2.13, which states that any group of order n has a generating set of size at most $\log n$. Also, we don't need to guess a parenthesization (although we still use $O(\log^2 n)$ nondeterministic bits to guess the generating set). Therefore, the reduction will output $((G, g_1, S), \dots, (G, g_n, S))$, and the proof concludes with the fact that

$$\text{rank}(G) \leq k \iff \bigwedge_{i=1}^n g_i \in \langle S \rangle. \quad \square$$

Theorem 4.2.

1. MAGMA RANK *is in* NP.
2. SEMIGROUP RANK *is in* NP.
3. QUASIGROUP RANK *is in* $\beta_2\text{FOLL}$.
4. GROUP RANK *is in* $\beta_2\text{FOLL} \cap \beta_2\text{L}$.

Proof. Follows from Lemma 2.1 and Lemma 4.1, along with

1. Lemma 3.1 for magmas,
2. Lemma 3.2 for semigroups,
3. Lemma 3.3 for quasigroups,
4. Lemma 3.5 for groups.

For groups the problem is also in $\beta_2\text{FOLL}$ since a group is a quasigroup. \square

These reductions can be generalized to the problem of computing the size of a minimum generating set for an arbitrary subset of the magma elements. The GENERALIZED MAGMA RANK problem is defined as follows (and there are analogous problems for the other algebraic structures). Given a magma G as a Cayley table, a finite set T of magma elements, and a natural number k in unary, decide whether there is a set S of size at most k such that $S \subseteq T \subseteq \langle S \rangle$. MAGMA RANK occurs as a special case when choosing $T = G$. Still, this problem reduces to the appropriate membership problem by the same reduction as in Lemma 4.1: nondeterministically choose a subset S of T with $|S| \leq k$, then decide whether each element of T is in $\langle S \rangle$.

We can reprove [6, Theorem 3.4] using this strategy as well. The alternate proof is a reduction from QUASIGROUP ISOMORPHISM to the join of *two* languages, CUBE MEMBERSHIP and PRODUCT EQUALITY. The latter is the problem of deciding whether two parenthesized products are equal according to the quasigroup operation given as a Cayley table. If the parenthesization is of depth $O(\log \log n)$, this problem is in FOLL by Lemma 2.6.

Theorem 4.3 ([6, Theorem 3.4]). QUASIGROUP ISOMORPHISM *is in* $\beta_2\text{FOLL}$.

Proof. This is a brief overview of the alternate proof. We will show a $\beta_2\text{AC}^0$ conjunctive normal form truth-table reduction from QUASIGROUP ISOMORPHISM to the join of CUBE MEMBERSHIP and PRODUCT EQUALITY, both of which are in FOLL. Assuming this reduction exists, we conclude using a proof similar to that of Lemma 2.1 that QUASIGROUP ISOMORPHISM is in $\beta_2\text{FOLL}$.

The reduction first guesses two cube generating sequences, \mathbf{g} for G and \mathbf{h} for H , both of length $O(\log n)$ and a parenthesization P of depth $O(\log \log n)$, then outputs the conjunction of the following queries.

$$\bigwedge_{g \in G} g \in \text{cube}_P(\mathbf{g}) \quad (1)$$

$$\bigwedge_{h \in H} h \in \text{cube}_P(\mathbf{h}) \quad (2)$$

$$\bigwedge_{\epsilon, \eta, \nu \in \{0,1\}^k} (P(\mathbf{g}^\epsilon) = P(\mathbf{g}^\eta) \cdot P(\mathbf{g}^\nu) \iff P(\mathbf{h}^\epsilon) = P(\mathbf{h}^\eta) \cdot P(\mathbf{h}^\nu)) \quad (3)$$

The first two formulas ensure that \mathbf{g} and \mathbf{h} are cube generating sequences for G and H , respectively. In the third formula, if $\mathbf{g} = (g_0, g_1, \dots, g_k)$, then \mathbf{g}^ϵ denotes $(g_0, g_1^{\epsilon_1}, \dots, g_k^{\epsilon_k})$. This formula checks that the bijection $g_i \mapsto h_i$ is a homomorphism.

Each of the first two formulas comprises n conjunctive queries to CUBE MEMBERSHIP. The last formula comprises a polynomial number of queries in conjunctive normal form to PRODUCT EQUALITY. Thus we have the required reduction. \square

We conclude this section with a few observations about Theorem 4.2. First, in this proof, we did not use the reduction from QUASIGROUP RANK to BOUNDED SUBQUASIGROUP MEMBERSHIP, because the closure of $\beta_2\text{FOLL}$ under $\beta_2\text{AC}^0$ conjunctive truth-table reductions is NFOLL, that is, FOLL with a polynomial amount of nondeterminism, whereas the closure of FOLL under the same reductions is $\beta_2\text{FOLL}$, a subset of NFOLL.

Second, a slight generalization of [21, Theorem 7] already proves that MAGMA RANK is in (and complete for) the class of problems decidable by a polynomial time Turing machine with $O(n \log n)$ nondeterministic bits. We have nevertheless included the fact that MAGMA RANK is in NP to highlight the general strategy for proving these upper bounds for each class of algebraic structure.

Third, we can almost show a reduction in the opposite direction of Lemma 4.1. The SUBMAGMA RANK problem (a search problem) is defined as follows. Given the Cayley table of a magma and a set of magma elements S , output the rank of $\langle S \rangle$. (The SUBMAGMA RANK problem is more general than the MAGMA RANK problem: the latter reduces to the former by choosing $S = G$.)

Proposition 4.4. *SUBMAGMA MEMBERSHIP reduces to the SUBMAGMA RANK function by a nonadaptive AC^0 Turing reduction making exactly two queries.*

Proof. We know that for any magma G , any magma element h , and any subset of magma elements S ,

$$h \in \langle S \rangle \iff \text{rank}(\langle S \rangle) = \text{rank}(\langle S \cup \{h\} \rangle).$$

(This is analogous to the corresponding situation in linear algebra: a vector h is in the span of a set of vectors S exactly when the rank of S does not increase when h is added to S .) Thus the generator of the reduction is the function $(G, h, S) \mapsto ((G, S), (G, S \cup \{h\}))$ and the evaluator of the reduction compares $\text{rank}(S)$ to $\text{rank}(\langle S \cup \{h\} \rangle)$ for equality. \square

However, this reduction is not satisfying, because the SUBMAGMA RANK is essentially the MAGMA RANK problem when the input is provided as a set of generators instead of as a Cayley table. As stated in the reduction, this representation may be exponentially smaller than the Cayley table representation.

Fourth, although the precise relationship between FOLL and L is unknown, FOLL does not contain any class containing the PARITY problem. Since PARITY is in L, we know FOLL does not contain L. Stated in a slightly more general way, FOLL cannot be hard under AC^0 many-one reductions for any complexity class that contains PARITY [5, Proposition 2.1]. This is true even when the circuit is augmented with a polylogarithmic number of nondeterministic bits [6, Section 4]. This gives an immediate improvement to the upper bound of the QUASIGROUP RANK problem.

Theorem 4.5. *QUASIGROUP RANK is not hard under AC^0 many-one reductions for any complexity class containing PARITY.*

Specifically, QUASIGROUP RANK is not hard for any of the classes in the inclusion chain

$$\text{ACC}^0 \subseteq \text{TC}^0 \subseteq \text{NC}^1 \subseteq \text{L} \subseteq \text{NL} \subseteq (\text{LOGCFL} \cup \text{DET}).$$

Finally, we consider whether there is a randomized logarithmic space algorithm for GROUP RANK, which would immediately improve the upper bound of Theorem 4.2. Let RL be the class of languages L for which there is a deterministic Turing machine with a two-way read-only tape for the input string, a one-way read-only tape for the random bits, and a two-way read-write work tape in which only $O(\log n)$ cells are used, such that $x \in L$ implies that at least half the binary strings r of polynomial length cause the machine to accept on input x and random bits r and $x \notin L$ implies that none of the binary strings r cause the machine to accept. Let $\rho_2\text{L}$ be the class of languages for which the tape containing the random bits is *two-way* read-only tape, and for which the length of the binary string r is $O(\log^2 n)$. By [20, Corollary 1] (see also [19]), $\text{RL} \subseteq \rho_2\text{L}$, and by the definitions, $\rho_2\text{L} \subseteq \beta_2\text{L}$. Now our question is whether GROUP RANK is in $\rho_2\text{L}$, or even in RL.

5 Computation of ring rank

The previous section provides an upper bound on the computational complexity of computing the rank of a group. Is it interesting to ask about the “rank” of rings or fields, and what does rank mean for these algebraic structures? A ring comprises an additive group and a multiplicative monoid with an additional distributivity property relating the two. We represent it as a pair of Cayley tables, one for the group and one for the monoid.³

Since our purpose for asking questions about the rank of an algebraic structure is to determine the smallest number of elements required to generate all other elements, we define the *rank of a ring* R to be the minimum cardinality of a set S such that $\langle S \rangle = R$, where $\langle S \rangle$ is the closure of S under *both* addition and multiplication. The rank of a ring is bounded above by the minimum of the rank of its additive group and the rank of its multiplicative monoid, but there are finite rings that have strictly smaller rank; see Example 5.1. We believe it enlightening to show how to use the same strategy used to determine the complexity of MAGMA RANK on RING RANK. This section shows how to reduce the rank problem to the membership problem, then shows an upper bound on the complexity of the membership problem.

Theorem 5.4 below yields an upper bound of $\beta_2\text{AC}^1$ for computing the rank of a ring given as a pair of Cayley tables. The hardness of this problem lies in computing the rank with respect to the underlying monoid (which is just a semigroup with an identity element, so the worst-case complexity of the rank problem for monoids and for semigroups is the same). Improving the algorithm for computing the rank of a monoid (or semigroup) will immediately improve the algorithm we propose in this section for computing the rank of a ring.

Not all rings have an interesting rank problem. In the special case that the ring is a finite domain (that is, it has no nontrivial zero divisors), then by Wedderburn’s little theorem [18, Theorem 3 § 11.1], the ring is a finite field. The multiplicative group of any finite field is cyclic [18, Theorem 7 § 6.4], so the rank of a finite field, and hence any finite domain, is one and the computational problem is uninteresting. But in general, for arbitrary (commutative or non-commutative) rings, the problem has nontrivial complexity. Even for small commutative rings, the rank of the ring can be strictly smaller than the rank of the group and of the monoid.

Example 5.1 ([24]). The ring $(\mathbb{Z}/2\mathbb{Z})^3$, with both addition and multiplication defined componentwise, has rank strictly smaller than both the rank of its underlying group and the rank of its underlying monoid.

The group is the elementary abelian 2-group of order 2^3 , and so has rank three by Example 2.9. The monoid has rank four via the generating set $\{(0, 1, 1), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}$. This set is a generating set because the

³ The underlying additive group of a finite ring can also be represented as the direct sum of a finite number of cyclic groups of prime power order by the fundamental theorem of finite abelian groups. This is the representation chosen in, for example, [2] and [16] when designing algorithms for finite rings.

product of any pair of the first three elements yields the elements $(0,0,1)$, $(0,1,0)$, and $(1,0,0)$, and the product of any pair of those three yields the remaining element, $(0,0,0)$. No smaller set can generate the entire monoid, since the identity generates only itself and no other element generates it, and removing any one of the first three makes it impossible to generate an element with a zero in the same coordinate as in the removed element. Thus the rank of the monoid is four.

However, the rank of the ring is at most two. Starting with the generating set $\{(0,1,1), (1,1,0)\}$ we have

$$\begin{aligned}(0,1,1) + (1,1,0) &= (1,0,1), \\ (0,1,1) \cdot (1,1,0) &= (0,1,0), \\ (1,0,1) + (0,1,0) &= (1,1,1).\end{aligned}$$

At this point, we have generated each element in the generating set for the monoid, so we can generate all other elements. \square

In order to compute the rank of the ring, we reduce the problem to the corresponding membership problem as in section 4.

Lemma 5.2. $\text{RING RANK} \leq_{\text{ctt}}^{\beta_2 \text{AC}^0} \text{SUBRING MEMBERSHIP}$.

Proof. The reduction is identical to the ones in Lemma 4.1, and uses only $O(\log^2 n)$ bits of nondeterminism because any generating set must generate the additive group, and the additive group has a generating set of size at most $\log n$ by Lemma 2.13. \square

Lemma 5.3. $\text{SUBRING MEMBERSHIP} \leq_m^L \text{DIRECTED PATH}$.

Proof. Given a ring R (expressed as two Cayley tables, one for addition and one for multiplication), a ring element r , and a set of ring elements S , construct a directed, labeled graph as follows. The set of vertices is the set of all ring elements. There is an edge from x to y labeled (a,b) if $xa - b = y$. Let G be the subgraph induced by edges labeled by pairs (a,b) where both a and b are elements of S . Output the subgraph G , the source vertex 1, and the target vertex r .

The choice of edges (x,y) where $xa - b = y$ for some a and b in S deserves some justification. First, choosing the subgraph induced by edges of the form $(a,0)$ yields the Cayley graph of the ring's underlying multiplicative monoid. Similarly, choosing the subgraph induced by edges of the form $(1,b)$ yields the Cayley graph of the ring's underlying additive group. Second, the subring test states that a subset of a ring is a subring if it is closed under multiplication and subtraction and contains the identity element. Thus the transitive closure of the vertex representing the multiplicative identity under edges labeled with elements from the subset S is guaranteed to be a subring.

Looping over each pair of ring elements (x,y) and each pair (a,b) requires $O(\log n)$ space for a ring with n elements. Deciding whether to add a labeled

edge from x to y requires a constant number of Cayley table lookups, again requiring $O(\log n)$ space. Thus the reduction is computable in logarithmic space.

To prove correctness, suppose r is in the subring generated by S . One way to see that there is a path from 1 to r is to consider the sequence of multiplications and additions that produce r from 1. Let (c_1, \dots, c_n) be the sequence of ring elements and $(*_1, \dots, *_n)$ be the finite sequence of ring operations, each one either an addition or a multiplication, such that $1 *_1 c_1 \cdots *_n c_n = r$. The sequences must be finite because the cardinality of the ring is finite. Then the path from vertex 1 to vertex r is the sequence of edges $((a_1, b_1), \dots, (a_n, b_n))$, where (a_i, b_i) is $(c_i, 0)$ if $*_i$ is multiplication or $(1, -c_i)$ if $*_i$ is addition, for each $i \in \{1, \dots, n\}$.

For the converse, suppose there is a path from 1 to r of length k in the graph G , where $k \leq n$. Let $((a_1, b_1), \dots, (a_k, b_k))$ be the labels along the edges of that path. To facilitate the closed-form representation of r , let $b_0 = -1$ and $a_{k+1} = 1$. By construction of the graph G , we know $r = (\cdots ((1a_1 - b_1)a_2 - b_2) \cdots)a_k - b_k$, or more concisely,

$$r = - \sum_{i=0}^k \left[b_i \prod_{j=i+1}^{k+1} a_j \right].$$

As stated previously, the transitive closure of 1 in G is the subring generated by S . By the formula above, r is in the transitive closure of S , so it is in the subring generated by S . \square

Theorem 5.4. RING RANK is in $\beta_2\text{AC}^1$.

Proof. DIRECTED PATH is in NL, and NL is closed under logarithmic space many-one reductions, so SUBRING MEMBERSHIP is in NL, by Lemma 5.3. Since $\text{NL} \subseteq \text{AC}^1$, there is a $\beta_2\text{AC}^0$ conjunctive truth-table reduction from RING RANK to a problem in AC^1 . Thus, by a proof similar to that of Lemma 2.1, we have RING RANK in $\beta_2\text{AC}^1$. \square

Can the complexity of SUBRING MEMBERSHIP be reduced, thereby reducing the complexity of RING RANK? One way of showing that it cannot would be to prove it NL-complete. We conjecture it is by a reduction from DIRECTED PATH to SUBRING MEMBERSHIP, but see no obvious approach. The strategy from [16, Theorem 4.4], which reduces the isomorphism problem for graphs to that for rings, does not seem directly applicable, since the reduction only maintains adjacency, not connectedness.

On the other hand, if SUBRING MEMBERSHIP reduces to SUBGROUP MEMBERSHIP by a sufficiently tight reduction, then we could improve the upper bound for RING RANK to $\beta_2\text{L}$. However, such a reduction seems unlikely, since access to both addition and multiplication should allow more ring elements to be generated from a given set than access to addition alone. The converse reduction seems unlikely as well, since an arbitrary group is not necessarily abelian, and a non-abelian group does not admit a ring structure. Even for abelian groups, the previous concern about access to both addition and multiplication applies. We

conjecture that the two problems are incomparable with respect to many-one reductions of sufficiently low complexity.

6 Computing other types of ranks

The “rank” of a vector space is the number of vectors in any basis of the space. In this setting, the vectors must not only span (or “generate”) the space, but also be *linearly independent*. The rank as defined and used in the previous sections have no such independence requirement. However, we could use other definitions of rank for groups, semigroups, etc., that do require independence in some way. [11, 12] define five such ranks for semigroups. We apply the same analysis as in section 4 with the new definition to get some similar results; this section summarizes those similarities and differences.

Theorem 6.3 shows that for groups, most other rank definitions yield limited nondeterminism (or conondeterminism) algorithms with simple verifiers, similar to that of Theorem 4.2. The limited nondeterminism lens is capable of improving algorithms for computational problems whose conditions seem even more strict than that of the standard rank problem. One definition of rank, the large rank, seems unlikely to be solvable with limited nondeterminism, and we conjecture this problem is NP-complete.

Until now we have considered the most common definition of rank, the minimum cardinality of a generating set, but there are others. Here are the five common definitions of rank, as described in [11, 12]. For any magma G ,

$$\begin{aligned}\text{rank}_L(G) &= \min\{k \mid \text{each subset of cardinality } k \text{ is a generating set}\} \\ \text{rank}_u(G) &= \max\{|S| \mid S \subseteq G \text{ and } S \text{ is an independent set}\} \\ \text{rank}_i(G) &= \max\{|S| \mid S \subseteq G \text{ and } S \text{ is an independent generating set}\} \\ \text{rank}_\ell(G) &= \min\{|S| \mid S \subseteq G \text{ and } S \text{ is a generating set}\} \\ \text{rank}_s(G) &= \max\{k \mid \text{each subset of cardinality } k \text{ is an independent set}\}\end{aligned}$$

These are called *large rank*, *upper rank*, *intermediate rank*, *lower rank*, and *small rank*, respectively. The lower rank is the notion of rank discussed in the previous sections of the paper.

Proposition 6.1. *For each finite magma G ,*

$$\text{rank}_s(G) \leq \text{rank}_\ell(G) \leq \text{rank}_i(G) \leq \text{rank}_u(G) \leq \text{rank}_L(G).$$

Proof. [12] states this chain of inequalities for semigroups; we show that it holds for finite magmas as well.

The inequalities $\text{rank}_\ell(G) \leq \text{rank}_i(G) \leq \text{rank}_u(G)$ follow from the fact that the collection of independent generating sets is the intersection of the collection of independent sets and the collection of generating sets.

Next we prove that $\text{rank}_s(G) \leq \text{rank}_\ell(G)$. If $\text{rank}_\ell(G) = |G|$, then we are done since $\text{rank}_s(G)$ must be bounded above by $|G|$ by definition.⁴ Thus

⁴A semigroup G with $\text{rank}_\ell(G) = |G|$ is sometimes called a “royal semigroup” (because it has the highest possible rank) [7].

it suffices to consider only magmas with $\text{rank}_\ell(G) < |G|$. Assume with the intention of producing a contradiction that $\text{rank}_s(G) > \text{rank}_\ell(G)$. Let S be the generating set of cardinality $\text{rank}_\ell(G)$. Let $x \in G \setminus S$, which exists because $\text{rank}_\ell(G) < |G|$. Now $S \cup \{x\}$ is a generating set of cardinality at most $\text{rank}_s(G)$. By definition of $\text{rank}_s(G)$, this means $S \cup \{x\}$ is independent. But $x \in \langle S \rangle$ since S generates G , so S cannot be independent. This is a contradiction, therefore $\text{rank}_s(G) \leq \text{rank}_\ell(G)$.

Finally we prove that $\text{rank}_u(G) \leq \text{rank}_L(G)$. If $\text{rank}_L(G) = |G|$, then G is the smallest generating set, so $\text{rank}_\ell(G) = \text{rank}_L(G)$, which subsumes the equality $\text{rank}_u(G) = \text{rank}_L(G)$. Thus it suffices to consider only magmas with $\text{rank}_L(G) < |G|$. Assume with the intention of producing a contradiction that $\text{rank}_u(G) > \text{rank}_L(G)$. Let S be the independent set of cardinality $\text{rank}_u(G)$ and let x be an arbitrary element of S . Then the set $S \setminus \{x\}$ has cardinality at least $\text{rank}_L(G)$, so it is a generating set for G . This means $x \in \langle S \setminus \{x\} \rangle$, a contradiction with the hypothesis that S is independent. Therefore, $\text{rank}_u(G) \leq \text{rank}_L(G)$. \square

The framework of section 4 provides a simple way of determining the complexity of computing these functions. The decision problems corresponding to these rank functions are defined as follows.

$$\begin{aligned} \text{MAGMA LARGE RANK} &= \{(G, k) \mid \text{rank}_L(G) \leq k\} \\ \text{MAGMA UPPER RANK} &= \{(G, k) \mid \text{rank}_u(G) \geq k\} \\ \text{MAGMA INTERMEDIATE RANK} &= \{(G, k) \mid \text{rank}_i(G) \geq k\} \\ \text{MAGMA LOWER RANK} &= \{(G, k) \mid \text{rank}_\ell(G) \leq k\} \\ \text{MAGMA SMALL RANK} &= \{(G, k) \mid \text{rank}_s(G) \geq k\} \end{aligned}$$

Some of these are maximization problems and some are minimization problems, depending on whether the rank is a minimum or a maximum. The problems are defined similarly for the other algebraic structures.

We can construct nondeterministic or conondeterministic reductions as follows. Let G^k denote the collection of subsets of G of cardinality k .

$$\begin{aligned} \text{rank}_L(G) \leq k &\iff \forall S \subseteq G^k : S \text{ is a generating set} \\ \text{rank}_u(G) \geq k &\iff \exists S \subseteq G^k : S \text{ is independent} \\ \text{rank}_i(G) \geq k &\iff \exists S \subseteq G^k : S \text{ is an independent generating set} \\ \text{rank}_\ell(G) \leq k &\iff \exists S \subseteq G^k : S \text{ is a generating set} \\ \text{rank}_s(G) \geq k &\iff \forall S \subseteq G^k : S \text{ is an independent set} \end{aligned}$$

Since S is an independent set exactly when $x \notin \langle S \setminus \{x\} \rangle$ for each x and S is a generating set for G exactly when $g \in \langle S \rangle$ for each g in G , these reductions are

nondeterministic or conondeterministic truth-table reductions.

$$\begin{aligned}
\text{rank}_L(G) \leq k &\iff \forall S \subseteq G^k: \bigwedge_{g \in G} g \in \langle S \rangle \\
\text{rank}_u(G) \geq k &\iff \exists S \subseteq G^k: \bigwedge_{x \in S} x \notin \langle S \setminus \{x\} \rangle \\
\text{rank}_i(G) \geq k &\iff \exists S \subseteq G^k: \bigwedge_{g \in G} g \in \langle S \rangle \wedge \bigwedge_{x \in S} x \notin \langle S \setminus \{x\} \rangle \\
\text{rank}_\ell(G) \leq k &\iff \exists S \subseteq G^k: \bigwedge_{g \in G} g \in \langle S \rangle \\
\text{rank}_s(G) \geq k &\iff \forall S \subseteq G^k: \bigwedge_{x \in S} x \notin \langle S \setminus \{x\} \rangle
\end{aligned}$$

By Lemma 2.11, the upper rank of any finite group is $\log n$, so the small, lower, and intermediate ranks have an upper bound of $\log n$ as well. (For large rank, however, it seems that any bound must depend on the prime factorization of n .)

For any language L , let \bar{L} denote the complement of L in Σ^* . For any bit b , let Lb denote the set $\{wb \mid w \in L, b \in \Sigma\}$. For any languages L_0 and L_1 , let $L_0 \oplus L_1$ denote the *join* of L_0 and L_1 , defined by $L_0 \oplus L_1 = L_0 0 \cup L_1 1$.

Lemma 6.2. *For brevity, let SM denote SUBGROUP MEMBERSHIP.*

1. GROUP LARGE RANK $\leq_{ctt}^{\text{coNAC}^0}$ SM.
2. GROUP UPPER RANK $\leq_{ctt}^{\beta_2 \text{AC}^0}$ $\overline{\text{SM}}$.
3. GROUP INTERMEDIATE RANK $\leq_{ctt}^{\beta_2 \text{AC}^0}$ SM.
4. GROUP LOWER RANK $\leq_{ctt}^{\beta_2 \text{AC}^0}$ $\text{SM} \oplus \overline{\text{SM}}$.
5. GROUP SMALL RANK $\leq_{ctt}^{\text{co}\beta_2 \text{AC}^0}$ $\overline{\text{SM}}$.

Similar reductions can be shown for the other algebraic structures.

The fact that \mathbf{L} is closed under complement yields the following upper bound for computing the various types of rank.

Theorem 6.3.

1. GROUP LARGE RANK *is in* coNP .
2. GROUP UPPER RANK *is in* $\beta_2 \mathbf{L}$.
3. GROUP INTERMEDIATE RANK *is in* $\beta_2 \mathbf{L}$.
4. GROUP LOWER RANK *is in* $\beta_2 \mathbf{L}$.
5. GROUP SMALL RANK *is in* $\text{co}\beta_2 \mathbf{L}$.

If the Immerman–Szelepcsényi theorem [13, 23], which proves $\mathbf{NL} = \text{coNL}$, can be adapted to show $\beta_2 \mathbf{L} = \text{co}\beta_2 \mathbf{L}$, then small rank problem can be decided in $\beta_2 \mathbf{L}$ as well.

References

- [1] V. Arvind and Jacobo Torán. “The Complexity of Quasigroup Isomorphism and the Minimum Generating Set Problem”. In: *Algorithms and Computation*. Ed. by Tetsuo Asano. Vol. 4288. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006, pp. 233–242. ISBN: 978-3-540-49694-6. DOI: 10.1007/11940128_25.
- [2] V. Arvind and T. C. Vijayaraghavan. “The Complexity of Solving Linear Equations over a Finite Ring”. In: *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science*. Ed. by Volker Diekert and Bruno Durand. STACS 2005. Springer Berlin / Heidelberg, 2005, pp. 472–484. ISBN: 978-3-540-31856-9. DOI: 10.1007/978-3-540-31856-9_39.
- [3] Jürgen Avenhaus and Klaus Madlener. “The Nielsen Reduction and P-Complete Problems in Free Groups”. In: *Theoretical Computer Science* 32 (1984), pp. 61–76. DOI: 10.1016/0304-3975(84)90024-0.
- [4] David Barrington and Pierre McKenzie. “Oracle branching programs and Logspace versus P”. In: *Mathematical Foundations of Computer Science 1989*. Ed. by Antoni Kreczmar and Grazyna Mirkowska. Vol. 379. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 1989, pp. 370–379. ISBN: 978-3-540-51486-2. DOI: 10.1007/3-540-51486-4_84.
- [5] David Mix Barrington et al. “On the Complexity of Some Problems on Groups Input as Multiplication Tables”. In: *Journal of Computer and System Sciences* 63.2 (2001), pp. 186–200. ISSN: 0022-0000. DOI: 10.1006/jcss.2001.1764.
- [6] Arkadev Chattopadhyay, Jacobo Torán, and Fabian Wagner. “Graph Isomorphism is Not AC^0 -Reducible to Group Isomorphism”. In: *ACM Transactions on Computation Theory* 5.4 (Nov. 2013), 13:1–13:13. ISSN: 1942-3454. DOI: 10.1145/2540088.
- [7] Emilia Giraldes and John M. Howie. “Semigroups of high rank”. In: *Proceedings of the Edinburgh Mathematical Society*. 2nd ser. 28 (1 Feb. 1985), pp. 13–34. ISSN: 1464-3839. DOI: 10.1017/S0013091500003163.
- [8] Georg Gottlob. “Deciding Monotone Duality and Identifying Frequent Itemsets in Quadratic Logspace”. In: *Proceedings of the 32nd Symposium on Principles of Database Systems*. PODS ’13. New York, New York, USA: ACM, 2013, pp. 25–36. ISBN: 978-1-4503-2066-5. DOI: 10.1145/2463664.2463673.
- [9] Robert D. Gray. “The minimal number of generators of a finite semigroup”. In: *Semigroup Forum* 89.1 (2014), pp. 135–154. ISSN: 0037-1912. DOI: 10.1007/s00233-013-9521-8.
- [10] Raymond Greenlaw, H. James Hoover, and Walter L. Ruzzo. *Limits to Parallel Computation: P-completeness Theory*. New York: Oxford University Press, 1995.

- [11] John M. Howie and Maria Isabel Marques Ribeiro. “Rank properties in finite semigroups”. In: *Communications in Algebra* 27 (11 1999), pp. 5333–5347. ISSN: 0092-7872. DOI: 10.1080/00927879908826758.
- [12] John M. Howie and Maria Isabel Marques Ribeiro. “Rank properties in finite semigroups II: The Small Rank and the Large Rank”. In: *Southeast Asian Bulletin of Mathematics* 24 (2 May 2000), pp. 231–237. ISSN: 0129-2021. DOI: 10.1007/s10012-000-0231-2.
- [13] Neil Immerman. “Nondeterministic Space is Closed under Complementation”. In: *SIAM Journal on Computing* 17.5 (1988), pp. 935–938. DOI: 10.1137/0217058.
- [14] Neil D. Jones and William T. Laaser. “Complete problems for deterministic polynomial time”. In: *Theoretical Computer Science* 3.1 (1976), pp. 105–117. ISSN: 0304-3975. DOI: 10.1016/0304-3975(76)90068-2.
- [15] Neil D. Jones, Y. Edmund Lien, and William T. Laaser. “New Problems Complete for Nondeterministic Log Space”. English. In: *Mathematical Systems Theory* 10.1 (1976), pp. 1–17. ISSN: 0025-5661. DOI: 10.1007/BF01683259.
- [16] Neeraj Kayal and Nitin Saxena. “Complexity of Ring Morphism Problems”. In: *computational complexity* 15.4 (2006), pp. 342–390. ISSN: 1420-8954. DOI: 10.1007/s00037-007-0219-8.
- [17] Richard Lipton, Lawrence Snyder, and Y. Zalcstein. *The Complexity of Word and Isomorphism Problems for Finite Groups*. Tech. rep. Yale University, Mar. 1977.
- [18] W. Keith Nicholson. *Introduction to Abstract Algebra*. Wiley, 2012. ISBN: 1118135350.
- [19] Noam Nisan. “On read-once vs. multiple access to randomness in logspace”. In: *Structure in Complexity Theory Conference, 1990, Proceedings., Fifth Annual*. IEEE. July 1990, pp. 179–184. DOI: 10.1109/SCT.1990.113966.
- [20] Noam Nisan. “Pseudorandom generators for space-bounded computation”. In: *Combinatorica* 12.4 (1992), pp. 449–461. ISSN: 0209-9683. DOI: 10.1007/BF01305237.
- [21] Christos H. Papadimitriou and Mihalis Yannakakis. “On Limited Nondeterminism and the Complexity of the V-C Dimension”. In: *Journal of Computer and System Sciences* 53.2 (1996), pp. 161–170. ISSN: 0022-0000. DOI: 10.1006/jcss.1996.0058.
- [22] Omer Reingold. “Undirected connectivity in log-space”. In: *Journal of the ACM* 55.4 (Sept. 2008), 17:1–17:24. ISSN: 0004-5411. DOI: 10.1145/1391289.1391291.
- [23] Róbert Szelepcsényi. “The Method of Forced Enumeration for Nondeterministic Automata”. In: *Acta Informatica* 26.3 (1988), pp. 279–284. ISSN: 0001-5903. DOI: 10.1007/BF00299636.

- [24] Pierre-Guy Plamondon (<http://math.stackexchange.com/users/110831/pierre-guy-plamondon>). *Is there a finite ring whose rank is smaller than the rank of its group and its monoid?* Mathematics Stack Exchange. Feb. 2016. URL: <http://math.stackexchange.com/q/1649784>.